LabVIEW SEQUENCER - A SCRIPT DRIVEN, AUTOMATED TESTING ROUTINE

Amy Walsh
Measurement Technology Center (MTC)
Jet Propulsion Laboratory, California Institute of Technology

ABSTRACT

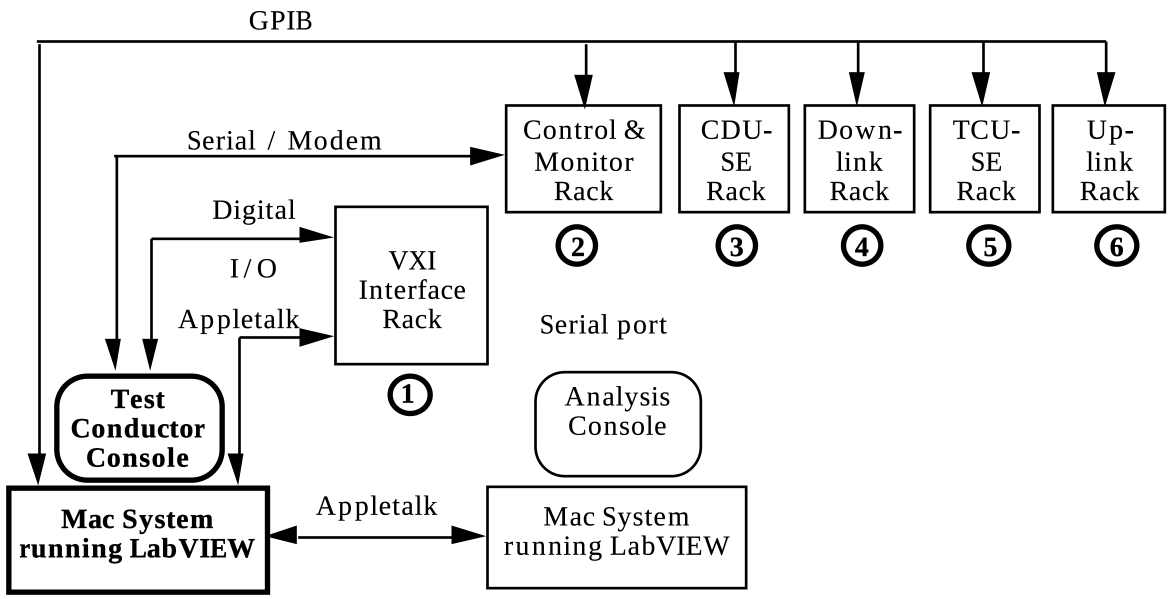
Testing of the Cassini Spacecraft Radio involves the control of and data acquisition from multiple and varied instruments and controllers. A system called the Sequencer was developed using LabVIEW 3.1 on a Macintosh Quadra 950 to facilitate acceptance testing of the Radio. The Sequencer is a script driven VI that commands the instruments, captures their data and configuration information, and stores the data for later analysis. Scripts are written by the test engineers using a language developed to make instrument setup easily learned and intuitive. Binary data is extracted into individual instrument tables that reside in a Microsoft Access database. A Visual Basic tool is used to retrieve data from the database for analysis and plotting.

INTRODUCTION

The Cassini Spacecraft is scheduled for launch in 1997. During its seven year cruise and the ensuing four year mission, the Radio Frequency Subsystem (RFS) onboard the spacecraft will be solely responsible for transmitting and receiving all communications signals between the spacecraft and Earth. For this reason, it is essential that the radio be thoroughly tested before launch. In order to test the Radio, an extensive collection of Support Equipment (SE) was put together to send communications signals to the Radio and receive and examine the signals returned by the Radio. In addition, the SE contains instruments with access to many of the voltage and current levels present in the Radio's circuits. Due to the number of instruments involved and the repetitive nature of configuring the instruments for test, computer control of the SE for testing was not only desirable, it was considered essential for completion of the testing on time and with a certainty that the instrument configuration and data gathering was done correctly.

REQUIREMENTS

Several issues were considered essential to the design of a program to execute Acceptance Test Procedures for the Radio. The test engineers wanted to minimize chances for operator error, keep track of the SE configuration at all times during a test, and be able to perform operations in any order. Verifying all of the specifications on the radio requires multiple SE and Radio configurations and the gathering of data from different instruments in different sequences. The SE instruments that must be configured include: around ten GPIB instruments, several PCs, serial instruments, and a distribution panel that requires digital I/O communication.



1 VXI Interface Rack
HP 8563E Spectrum analyzer
HP 8644B Synthesized signal generator

VXI Chassis
RD 2151 Microwave counter
VXIpc-030 (Macintosh, NI)
Multiplexer (RD1260-35)
Digital multimeter (HP E1410A)
Giga 58542 Power meter

2 Control & Monitor Rack
SSPS Solid state power supply
RFS SE Status panel

3 CDU-SE Rack
CDU-SE PC

5 TCU-SE Rack
TCU-SE PC
RTIU PC

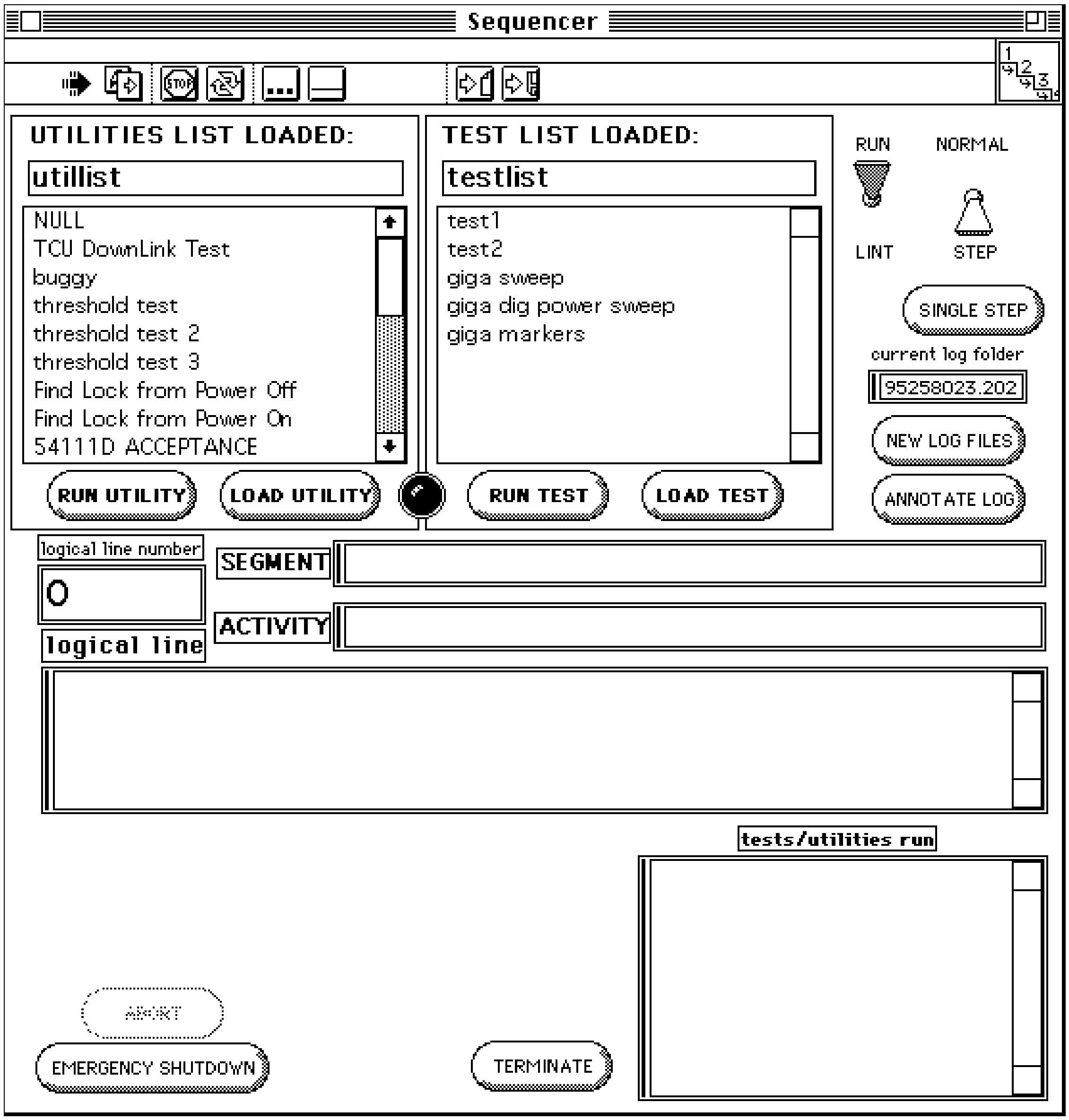
4 Downlink Rack
HP 8563E Spectrum analyzer
SA 930 Series telemetry receiver
HP 3586C Selective level meter
HP 5370B Time interval counter

6 Uplink Rack
HP3325A Synthesizer/Function generator
HP 54111D Digitizing oscilloscope
Gigatronics 7200 Synthesized sweeper

Radio Frequency Subsystem Support Equipment Layout

SOLUTION

The system designed to handle this set of requirements is the Sequencer, a LabVIEW 3.1 VI running on a Macintosh Quadra 950. The Quadra contains 2 NB-DIO-32F and 1 NB-DMA2800 boards. The Sequencer is a script-driven program that communicates with each of the Support Equipment instruments to perform all of the necessary instrument control, data acquisition, and data storage functions for acceptance testing of the Radio. The Sequencer contains over 250 subVIs that divide the tasks of reading and parsing the script file, identifying script commands, sending the appropriate instrument commands or operator interface messages, and logging the data to output files.

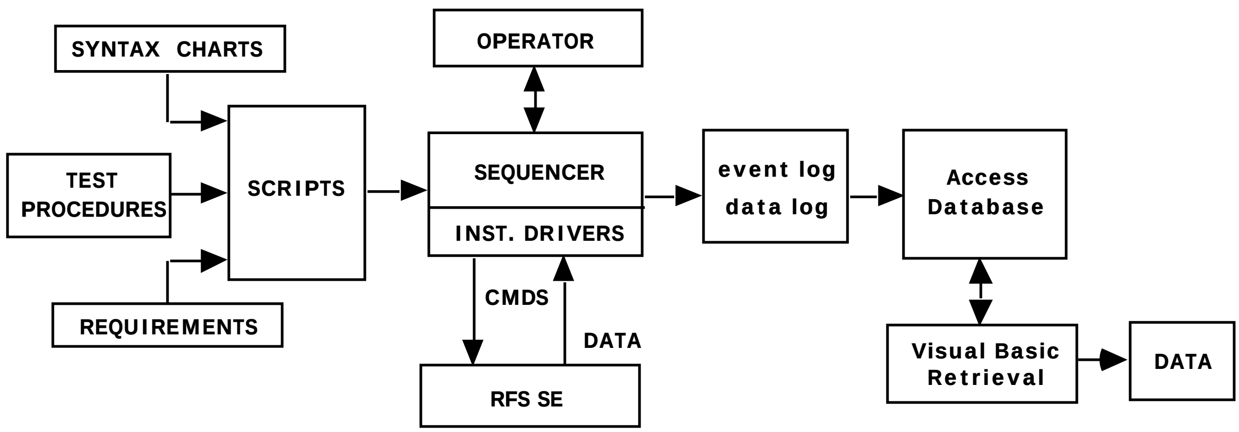


Sequencer Front Panel

Scripts are written by test engineers using a language created by the Sequencer instrument driver developers. Commands begin with a token describing the instrument to configure, followed by the specification of the command using one or more tokens. The tokens were chosen to describe the function or parameter they represent so that the test engineers would have a relatively easy time learning the script language. Feedback so far indicates that the test engineers find the script language easy to use.

The Sequencer provides a detailed record of its performance during the execution of a script by writing information out to two main log files. The event log is an ASCII text file that lists all the actions taken in executing a script. The data log is a binary file containing data records from the

SE instruments fully describing their configuration at the time the record is logged. A time stamp is included with each record. The binary file contains all of the data to be analyzed and plotted after a test. A process was developed to convert the binary file to a series of tab-delimited text files, each with information specific to a single instrument. This process is accomplished by another LabVIEW 3.1 VI. Since there are many tests and many records of data, a database was created to contain the data. Query and data retrieval programs were developed to help the engineers extract the data they need to analyze. The database used was Microsoft Access and the data retrieval program was written in Visual Basic 3.0.



Sequencer Input/Output Flow Diagram

A typical test for the Radio finds the threshold power at each point of a series of frequencies in the Radio's reception range. Three of the instruments involved in this test are a Gigatronics 7200 Synthesized Sweeper, the Telemetry Control Unit (TCU) of the Radio, and the Direct Access channels that gather voltage information from various sources within the Radio. The test involves locking the Radio to a signal, sweeping the Radio to the frequency of interest, and then dropping the power step by step until the Radio loses lock on the signal. A typical query for this test would be to look for all of the places in the Synthesized Sweeper records where the frequency is a certain value, and only retrieve records for which the TCU telemetry shows that the Radio is locked to the signal. Data from the TCU and Direct Access during the times that these two conditions are true will give a profile of how the Radio is behaving as the power of the uplink signal is fading away. This type of query can also be done for the rest of the frequency points of interest. The same data can also yield a plot of threshold power versus frequency over the range of frequencies tested. This data can be retrieved by performing a query that gets all of the data records from the Synthesized Sweeper at points immediately prior to the times the TCU telemetry data shows that the Radio has lost lock. The power and frequency values in these Synthesized Sweeper records will reveal the points along the threshold curve.

As the Sequencer was put into operation by the test engineers, they came up with additional requirements. One essential feature was that they needed the ability to loop until a specific instrument parameter reached or exceeded a certain value. To meet this requirement, script commands for looping and case statements were incorporated into the Sequencer system.

Another element that was added after the initial version was the ability to send the latest data for several instruments to another Macintosh Quadra 950 for display. This was accomplished by opening an Appletalk session with the second Quadra and sending data packets over which are parsed and displayed by VIs running on the second machine. The data transfer is relatively rapid and does not slow the Sequencer as much as having a display available on the computer running the Sequencer. With this data display capability on the second computer, the operator can view the latest data returned from a particular instrument in order to verify manually what a script file is keeping track of programmatically.

ACKNOWLEDGMENTS

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.